Quality Assurance and The SAS® System

Thalene T. Mallus
The National Institute of Mental Health

ABSTRACT
In an ideal research environment, all data would be entered and stored in the same format, from the same operating system, and would have no inaccuracies, duplicates, or data entry errors. Unfortunately, in the real research world, data managers and analysts are often faced with the challenge of extracting, combining, and cleaning up data that have been entered by a variety of personnel or obtained from multiple sites or repositories.

This paper describes the related concepts of data validation and data 'scrubbing', and reviews some general SAS System tools and techniques available to ensure the accuracy and consistency of existing data, as well as tools available to ensure the quality of new SAS data sets. General data quality assurance guidelines are also discussed.

INTRODUCTION
It is critical to the mission of the National Institutes of Health, the nation's largest medical research facility, to ensure the accuracy of data collected from its numerous research studies as well as the accuracy of data acquired and managed from a multitude of collaborative sources. When data integrity is violated, erosion of public trust can result, not only for a specific study, but for scientific research in general (Blumenstein et al.,1995).

Simply stated, data quality is a shared responsibility. Those responsible for data management must acquire and enter the data conscientiously to preserve its integrity. Quality assurance monitoring must continue throughout the entire research process beginning with data acquisition, through data transmittal and storage, and culminating in data analysis and reporting (Gassman et al., 1995).

In planning any research endeavor, one must assume that data requiring manual data entry will undoubtedly result in some degree of error (Blumenstein, 1993). Regardless of whether data is entered manually, scanned, or generated by laboratory equipment, computerization of data in no way guarantees data accuracy. Particularly for manually entered data, there is little assurance that the information is coded or entered in a consistent manner (Hobbs & Hawker, 1995).

Varied Personnel
In the true world of biomedical research, where private sector companies and government agencies are downsizing, many types of personnel with a variety of computer skills may be responsible for data entry and data management (e.g., volunteers, students, research assistants, secretaries, programmers, or doctoral level scientists).

When data management personnel is varied, the level of familiarity with the content of the data can differ greatly. The degree to which staff can detect or correct data anomalies is greatly influenced by their knowledge of the details and background of the scientific question(s) under study.

Varied Data Organization
In addition to differences in data management personnel, scientific data often come from a variety of sources. It is common practice for NIH scientists to collaborate with investigators from within NIH or with institutes or universities worldwide in an effort to share scientific information.

It is also quite common for different scientists to organize the same information in a variety of ways. Information collected at one site may be named, categorized, or stored in a completely different manner from identical data collected at another institute or university. (For example, one site may enter the variable SEX = M,F; another as SEX=1,2; yet another as GENDER= 0,1.)

All of these situations can lead to 'dirty' data, which can range from inconsistent variable names or values, to data that contains more serious anomalies such as keystroke errors, interpretation errors, or duplicate records. Any data that is incomplete or otherwise inaccurate can be considered 'dirty'.

All research studies should, at some level, implement quality assurance monitoring. Data originating on legacy systems (older mainframe systems) or from numerous investigators or sites are particularly vulnerable to a variety of data errors. Such data often need to be cleaned and organized into a single database system for efficient access and data reporting.

The purpose of this paper is to discuss the related concepts of data validation and data scrubbing as they relate to the SAS system and to provide general tips to users and new application developers on how the SAS system can assist in their goal towards accurate, reliable, and otherwise 'squeaky clean' data.

DATA VALIDATION vs. SCRUBBING
Data Validation
Data validation is the process of verifying that keyed, scanned, or transferred data are congruent with the original source. Usually a data entry application incorporates validation

procedures which detect anomalies such as the absence of required values, data coding and format errors, as well as range and consistency errors (Blumenstein, 1993).

Although visual data validation procedures are generally implemented at the completion of data entry, most automated data validation techniques are normally implemented interactively during data entry so that errors and inconsistencies can be corrected immediately (Blumenstein, 1993).

The process of data validation is extremely important for any research project, but even more so for medical research. It is crucial to the mission of NIH to minimize all possible data errors, since many questions under investigation deal with life threatening illnesses or serious medical conditions. Particularly for studies such as cancer or AIDS research, data integrity violations can be a serious consequence for patients dependent on new breakthroughs.

Further, when data are found to be invalid, re-analysis and re-interpretation of the data can induce substantial costs, often resulting in lost production time, and delays in publishing important findings (Blumenstein, 1993).

Data Scrubbing

Data scrubbing refers to a cleansing process implemented after the original data have been entered, but prior to the extraction of the data from another operating system or database (Bort, 1995).

In general, databases get ‘dirtier’ as the number of files and age of the data increases. Information in older databases can have any number of inconsistencies as compared with current database counterparts.

Older data may have values that are inconsistent with the current SAS data set into which it will be incorporated. Variable names between two databases may also be different, even though the variables are referring to identical information (e.g., SEX vs GENDER). Often, there are numerous misspelled names attached to the same patient or client identification number resulting in duplicate records for the same individual (Bort, 1995).

When routine database housekeeping is not completed, data integrity can be seriously compromised. Further, as the physical storage space of the database increases, a noticeable deterioration of performance can occur resulting in delays for the user (Blumenstein, 1993).

Because the SAS System offers a comprehensive set of integrated tools, it is equipped to master even the toughest data validation or scrubbing challenge. Many of the procedures and techniques described in this paper are available with Base SAS software and can be applied to either validation or scrubbing processes depending on the needs of the research study.

Although some of the same techniques can be used, it is important to distinguish between validation and scrubbing since data which have undergone a stringent validation process at the onset, often require little if any scrubbing later (Bort, 1995).

For those new to application development, this paper also describes some rudimentary validation examples from SAS/FSP® . Unfortunately, in a paper of limited length it is impossible to discuss each and every data verification technique in detail, so only a brief description of basic quality assurance concepts will be illustrated.

A DATA SET EXAMPLE

To illustrate the quality assurance techniques of data validation and data scrubbing, three sample data sets have been created from a National Institute of Mental Health (NIMH) study examining etiological factors in the relatives of schizophrenic adoptees. The Danish-American Provincial study involves a multitude of information gathered from comprehensive interviews and hospital records, with data ranging from general demographics to character traits and schizophrenic symptoms.

MANUAL vs. AUTOMATED VALIDATION

Although most current validation techniques rely to some degree on computerization, some techniques rely more heavily on manual-visual comparisons than others.

The CONTENTS Procedure

For example, Table 1 was formulated by extracting variable information from three separate PROC CONTENTS outputs. The code used to generate the information is:

```
libname statements ... ;
proc contents data=in1.MASTER position;
proc contents data=in2.VERIFY position;
proc contents data=in3.SCRUBME position;
run;
```

Table 1 displays three sample data sets with their corresponding variable names, type, length, and descriptions. All three data sets represent permanent SAS data sets. For this paper, the MASTER data set is assumed to be the primary database. The VERIFY data set is a duplicate keying of the MASTER data set, but with data entry errors. The SCRUBME data set represents fictitious data presumably entered at a collaborating site by an individual unfamiliar with the study standards.

As shown in Table 1, the MASTER and the VERIFY data sets both contain the same 19 variables, all of which are numeric, except for the SEX character variable. The SCRUBME data set is missing four variables (FRIENDS - AGELAST), and has a numeric variable GENDER, instead of the character variable SEX. SCRUBME also has non-standard variable names for [AFFECT], [ONSET], and [HOSP].

Although somewhat informative, PROC CONTENTS is limited in the amount of information it can provide. Any direct comparisons between data sets must be manually completed by the user.

The MEANS Procedure

Another method which provides a bit more information about the values in a data set is PROC MEANS (see code below). When the N, NMISS, MINIMUM, and MAXIMUM options are used with PROC MEANS, more detail regarding the numeric values can be obtained.

```
libname statements ... ;
proc means data=in1.MASTER n nmiss min max
                           maxdec=0;
*... repeat code for in2.VERIFY and
     in3.SCRUBME;
run;
```

When the columns containing the variable names are reviewed (see Output 1), we can see that MASTER and VERIFY have the same number of numeric variables with the same names and variable descriptions. Although the number of subjects are equal for the two data sets (N=53), this fact alone does not ensure that the VERIFY data contains the exact same cases as the MASTER data.

In fact, as the N (number of observations) and NMISS (number of missing observations) columns are examined, it appears as if the VERIFY data might contain some missing data. Since VERIFY was designed to be an exact duplicate of MASTER (a second keying of the data), we would expect the N and NMISS columns for the two data sets to be equal if all cases in both MASTER and VERIFY were entered correctly. Since the output of the MEANS procedure cannot tell us whether the errors reside in the MASTER or the VERIFY data, the specifics of any data entry errors cannot be easily ascertained.

Further, when the MINIMUM and MAXIMUM range columns are examined, we discover that the minimum range for the variable AGEONSET in VERIFY is 1. Although a possible value, it is most unlikely that an individual would actually have a psychiatric onset at the age of 1. Consequently, each value of AGEONSET would be suspect and each value would have to be verified so that any errors could be identified and corrected.

For small data sets these methods work quite well and are completely acceptable verification techniques. It quickly becomes apparent, however, that as the number of variables or data sets increase, manual methods soon become laborious and inefficient.

Further, manual or visual comparison methods rely heavily on the transient energy levels of the individual completing the comparison. Many factors can affect the accuracy of manual comparisons such as the appearance of the data, environmental factors, fatigue, boredom, or stress (Blumenstein, 1993).

AUTOMATED VALIDATION

One possible solution, if fiscal and temporal constraints allow, is to double key the data (or at least critical fields) and utilize an automated comparison program to determine the existence of discrepancies or data anomalies (Blumenstein, 1993). Although this method will not detect errors in interpretation (such as assigning the wrong diagnostic code), it does provide the user with a detailed report of both observation and variable discrepancies.

The COMPARE Procedure

In the SAS system, the COMPARE procedure effortlessly examines the values of two SAS data sets no matter how large. It also has the capability to compare the values of different variables within the same SAS data set.

Output 2 is a report generated by PROC COMPARE (see code below) which compares the VERIFY data set (the second keying of MASTER) to the MASTER data set (the primary database) for a single variable FINALDX.

```
libname statements ... ;
proc sort data=in1.MASTER out=master;
   by SUBJ;
proc sort data=in2.VERIFY out=verify;
   by SUBJ;
proc compare data=master compare=verify list;
   id SUBJ;
   var FINALDX;
run;
```

Output 2 illustrates that the information obtained by the LIST option of the COMPARE procedure is much more specific and detailed than the manual methods previously described. In the output, we discover that the VERIFY data set contains a duplicate observation for subject #1 and that subject #321 of MASTER is missing.

Finally we are presented with a value comparison table which indicates that four observations in the VERIFY data set have mismatched values (which appear to be inversion errors) for the diagnostic variable FINALDX. These four records can now easily be validated against the original source documents for the correct values, thereby substantially increasing our confidence in the validity of the MASTER data.

It should be noted that the PROC COMPARE LIST option, which lists all variables and observations found in any one data set, is only one of over 30 available options. PROC COMPARE is an extremely powerful procedure which can result in a tremendous amount of information about two data sets (see SAS Procedures Guide for more information).

SAS/FSP Data Entry Validation

Thus far, all of the techniques discussed have been validation techniques utilized at the *completion* of data entry. The SAS System also has the capability to customize data entry applications to include validation *during* data entry.

Both SAS/FSP® and SAS/AF® software have the capability to perform validation checks during data entry. SAS/AF is used to create user friendly applications in an interactive windowing environment. SAS/FSP is generally used to browse and edit permanent SAS data sets or external files.

Although browsing and editing are the primary functions of SAS/FSP, the product also has the capability to create customized data entry applications with tailored screens and built-in validation checks (Aster, 1994). Because our laboratory is licensed to use SAS/FSP, I will discuss some common automated validation checks available for that product.

SAS/FSEDIT Development Environment

SAS/FSP software contains four procedures:

1. PROC FSEDIT /FSBROWSE - displays and modifies records one observation at a time.
2. PROC FSVIEW - displays and modifies records in a table format.
3. PROC FSLETTER- allows the creation of text documents or form letters which use variables from a SAS data set.
4. PROC FSLIST- displays text files for browsing.

(Aster, 1994)

Of these procedures only FSEDIT/ FSBROWSE and FSVIEW contain development environments to create data entry and data presentation applications. Only the FSEDIT development environment will be discussed here.

In PROC FSEDIT the development environment is separate from the execution environment. To execute PROC FSEDIT, one need only issue the following code from the Program Editor of the SAS Display Manager:

```
libname statement ...;
proc fsedit data=in.SASFILE;
run;
```

The above code will provide the user with a default screen and will allow a permanent SAS data set to be searched and modified interactively. Aside from the default screen, customized screens can also be built that are more user friendly and can assist with data entry tasks. These customized data entry applications (see Chapter 8, SAS/FSP Software manual for more details) can be programmed to contain validation checks during data entry.

To enter the development environment for FSEDIT, issue the MODIFY command from the FSEDIT window. The MODIFY command opens the FSEDIT Menu.

Option 4 of the FSEDIT Menu, Assign Special Attributes to Fields, allows the programmer to assign specific field attributes to one or all of the variables in a data entry application. Some common field attributes are:

- INITIAL- provides an initial value to a field.
- MINIMUM - provides a minimum value to a field.
- MAXIMUM - provides a maximum value to a field.
- REQUIRED - controls whether a value must be entered in a field.
- ECOLOR - selects a text color when the value entered is invalid. (See Chapter 13, SAS/FSP software for more options).

Figure 1 illustrates a customized data entry screen for the MASTER data. For the variable PARANOIA, we observe that an invalid value was entered (9) which was greater than the MAXIMUM value allowed (5). The end user will not be allowed to continue to enter more data until a value within an acceptable range is entered (see error message, Figure 1).

Using SCL Within the FSEDIT Environment

Customized screen entries can also contain more sophisticated validations created by Screen Control Language (SCL). SCL is a general programming language similar to SAS which has a high level flow structure and distinct features related to windowing environments (Aster, 1994). Often a programmer develops logic checks where the value of one field is compared against the value in another field.

The following code is written in SCL and performs a logic check for the fields AGEONSET and AGE1HOSP. The rationale for the code is that the age of an individual's first psychiatric onset cannot be greater than the age of that patient's first psychiatric admission.

```
INIT:
return;
MAIN:
/*logic check for age onset vs age 1st hosp*/
 if (AGEONSET>AGE1HOSP) then do;
   erroron AGEONSET;
   _msg_='Invalid Data for AGEONSET or AGE1HOSP';
   cursor AGEONSET;
end;
return;
```

```
TERM:
return;
```

To execute this SCL code, it must first be compiled by the SAS system. Option 3, Edit Program Statements and Compile, must be chosen from the FSEDIT development menu. Figure 2 illustrates the results of an invalid entry for Age of Psychiatric Onset (AGEONSET). Again, the user must correct the invalid field before data entry can continue.

SCL can provide any number of sophisticated automated validations. Logic checks or cross validations are only one type. SCL can be used to display selection lists of field values, perform table lookup validations, create calculations using field values, or display messages or alarms when invalid values are entered. SCL is extensive and thorough. Any number of possible validations or verification rules are possible.

DATA SCRUBBING

Even the best data managers encounter situations where data are in need of a good scrubbing. Despite attempts to validate data beforehand, data managers are often plagued by data that have either been inadvertently mismanaged or precariously entered. Despite our best intentions, often collaborating sites do not communicate their data entry rules or formats until the data are already entered and ready for analysis. When it comes time to share the data, we often find that variable names or values are incompatible between two or more data sets.

Scrubbing techniques use many of the same Base SAS tools as validation techniques. For instance, PROC SORT can be used with the NODUP or NODUPKEY to eliminate duplicate observations or observations with duplicate by values. Likewise, statements which utilize character strings can be developed to locate and correct different spellings of the same patient or client name.

The MEANS Procedure Revisited

If we look back at the PROC MEANS Output 1, we notice that there are a number of significant differences between the MASTER and the SCRUBME data sets. Remember that the SCRUBME file represents data entered at a fictitious site by an individual unfamiliar with the study standards.

From Output 1 we learn that SCRUBME has no variable labels, has four missing variables (FRIENDS, TEASE, SHY, and AGELAST), has a number of different variables or variables with non-standard names (GENDER, AFFECT, ONSET, HOSP), and has at least one variable, DELUSION, with variable values in a suspicious range (1 to 5, instead of 0 to 1; with 0 indicating no delusions and 1 indicating some evidence of delusions).

We expect that the 14 cases in SCRUBME are all new observations to be concatenated to our primary database MASTER. However, we obtain no information from this output to confirm whether the 14 observations are unique or whether any of the cases are duplicates.

Visualizing Data

Once data anomalies are suspected, it is often useful to plot the data to check for any unusual patterns. Figure 3 displays comparison plots for the variable DELUSION in both the MASTER and SCRUBME data sets (see code below). As expected, the MASTER data set shows a parallel pattern with fewer cases having delusions (1). SCRUBME, on the other hand, shows most cases with evidence of delusions (1) and two cases with questionable values (3,5).

```
libname statements ... ;
proc plot data=in1.MASTER nomiss hpercent=30;
title 'Plot for Delusion in MASTER';
 plot SUBJ*DELUSION='M'/vaxis=0 to 950 by 100;
proc plot data=in1.SCRUBME nomiss hpercent=30;
title 'Plot for Delusion in SCRUBME';
 plot SUBJ*DELUSION='S'/vaxis=0 to 950 by 100;
run;
```

After examining the plots in Figure 3, a skilled investigator or a well-versed data manager would suspect (and verify with the collaborating site), that the (1) values for DELUSION in SCRUBME should really be (0) and the values (3,5) should actually be missing values.

Recoding with User Defined INFORMATS

Cody (1995) wrote a very useful paper explaining how data can be scrubbed by utilizing user defined SAS INFORMATS. To clean up the DELUSION variable in SCRUBME, PROC FORMAT with the INVALUE statement as well as the INPUT FUNCTION is used:

```
libname statements ... ;
proc format library=library;
 invalue delus 0=1 1=0 other=.;
data in.SCRUBME2;
 set in.SCRUBME;
 DELUSION=input(DELUSION,delus.);
run;
```

This example allows us to correct the invalid values of DELUSION in SCRUBME (1’s converted to 0’s), and at the same time, all values which are not (0) or (1) are automatically recoded to missing. Now if we plot the data from the corrected file (SCRUBME2) for DELUSION (see Figure 4), we find that the data are more to our expectations, indicating most cases have no delusions. Note that when no symbols are defined in PROC PLOT, SAS indicates one observation with an A, two with a B, three with a C, etc. When symbols are defined (as in Figure 3), it can be difficult to determine where coinciding data reside.

The COMPARE Procedure Revisited

Finally, the COMPARE procedure is also an excellent tool to provide information about variable name inconsistencies between SAS data sets. With the LISTVAR option, users can determine the number of variables in common as well as which variables are contained in one data set and not in another. If the ID option is used, PROC COMPARE will also provide information about the number of observations in common and whether any duplicate observations exist.

GENERAL QUALITY ASSURANCE TIPS

Although no data management system is perfect, there are some general guidelines which, once implemented, can result in more reliable, consistent data.

- Develop variable names which reflect the content of the data, and where possible, use character values (M,F) instead of numeric values (1,2) to assist users with data entry.
- Develop variable standards and use permanent SAS data sets, FORMATS, INFORMATS and Variable Labels to ensure consistency of variable names, content, and values between data sets and among collaborating sites.
- Assume that all manually entered data will have mistakes and program required values and validation checks into all applications.
- Automate routine checks to ensure data accuracy and completeness.
- Ensure the accuracy of data codes by providing the user with customized screens and/or selection lists.
- Double enter data and check a random sample or key fields to ensure that data entered accurately reflect source documentation.
- Where possible, use SAS procedures instead of self-developed code, and try to determine the most efficient procedure for the task at hand.

CONCLUSION

In today's research environments it is common to receive data in less than perfect condition. Use of a computer system does not guarantee proper classification or consistency, nor does it guarantee standardization of approach or accuracy (Hobbs & Hawker, 1995).

In validating data, one should try to utilize the most efficient combination of quality control measures available, based on the study size and design. Not all studies require sophisticated automated systems. Advanced computerized tools can be overkill for small studies where manual systems are just as effective (McFadden et al., 1995).

Given the paucity of publications about data management in clinical and epidemiologic research, and the lack of comparisons for different quality assurance methods, it would serve the scientific community well to include a brief description of any quality assurance techniques utilized for a specific study (just as scientific methodologies are currently included). It is important for readers to know which, if any, quality assurance methods are utilized, because without such elaborations, how can we be sure our findings reflect the best the data have to offer?

TRADEMARKS

REFERENCES

Aster R. *SAS Foundations from Installation to Operation,* New York: Windcrest / McGraw-Hill, 1994.

Blumenstein BA. Verifying keyed medical research data. *Statistics in Medicine*, 1993;12:1535-1542.

Blumenstein BA, James KE, Lind BK, Mitchell HE. Functions and organization of coordinating centers for multicenter studies. *Controlled Clinical Trials*. 1995; 16: 4S-29S.

Bort J. Scrubbing dirty data. Info World, 1995; 17: 1,57-58.

Cody R. Some clever things to do with user defined informats. *NESUG'95 Conference Proceedings.* Washington, DC, 1995.

Gassman JJ, Owen WW, Kuntz TE, Martin JP, Amoroso WP. Data quality assurance, monitoring, and reporting. *Controlled Clinical Trials*. 1995;16: 104S-136S.

Hobbs FDR, Hawker A. Computerised data collection: practicability and quality in selected general practices. *Family Practice*. 1995;12(2):221-226.

McFadden ET, LoPresti F, Bailey LR, Clarke E, Wilkins PC. Approaches to data management. *Controlled Clinical Trials*. 1995;16:30S-65S.

SAS Institute Inc. *SAS /FSP Software: Usage and Reference*, Version 6. Cary, NC: SAS Institute Inc, 1989.

SAS Institute Inc. *SAS Language: Reference, Version 6*. Cary, NC: SAS Institute Inc, 1990.

SAS Institute Inc. *SAS Procedures Guide,Version 6*. Cary, NC: SAS Institute Inc, 1990.

SAS Institute Inc. *SAS Programming Tips: A Guide to Efficient SAS Processing*. Cary, NC: SAS Institute Inc, 1990.

SAS Institute Inc. *SAS Screen Control Language: Usage, Version 6.* Cary, NC: SAS Institute Inc, 1990.

ACKNOWLEDGEMENTS

Loring Ingraham, Ph.D./ Psychologist/ NIH/NIMH
SAS Technical Support/SAS Institute Inc./ Cary, NC

THE AUTHOR MAY BE CONTACTED AT:

The Laboratory of Psychology and Psychopathology,
National Institute of Mental Health
Building 10, Room 4C110
10 CENTER DR MSC 1366
BETHESDA MD 20892-1366

Phone: (301) 496-7672 ext. 35
FAX: (301) 402-0921, E-Mail:Thalene_Mallus@nih.gov